Neural Networks - Image Recognition

In [1]:
```
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
```

In [2]:
```
import matplotlib.pyplot as  plt
%matplotlib inline
```

In [3]:
```
# 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
```

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.**

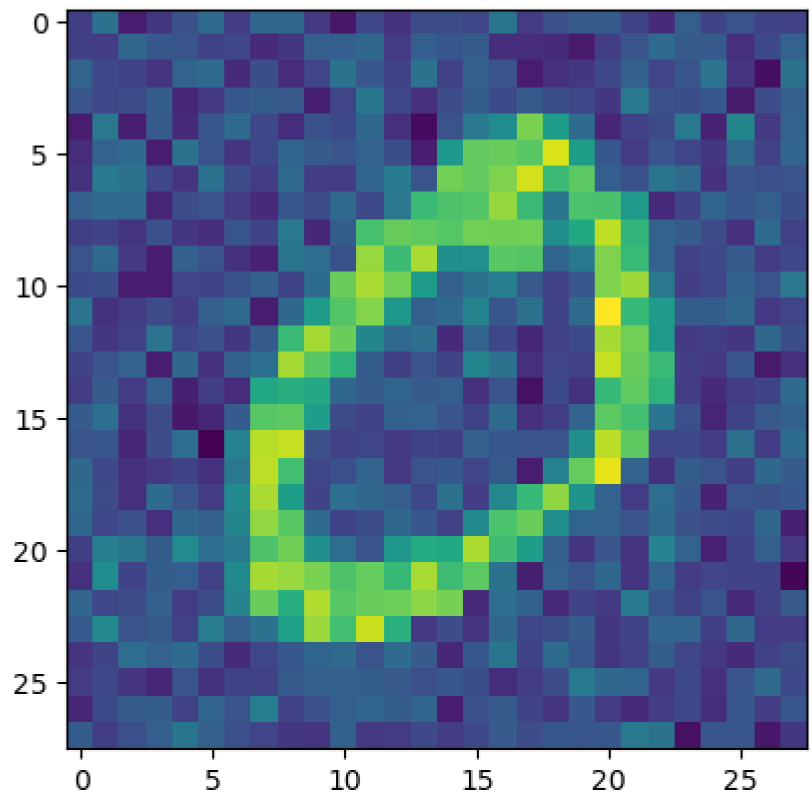
In [4]:
```
# Noise is added here

# Generate random noise with the same shape as the image
import numpy as np
mean = .5
stddev = .16
noise = np.random.normal(mean, stddev, x_train[1].shape)

# Add the noise to the image
image_noise_added = x_train[1] + noise
image_noise_added


# The max value of the noise should not grossly surpass 1.0
max(noise)
```

Out[4]:
```
1.000400027849817
```

In [5]:
```
# Max of noise is less than .1002; very slightly over 1
```

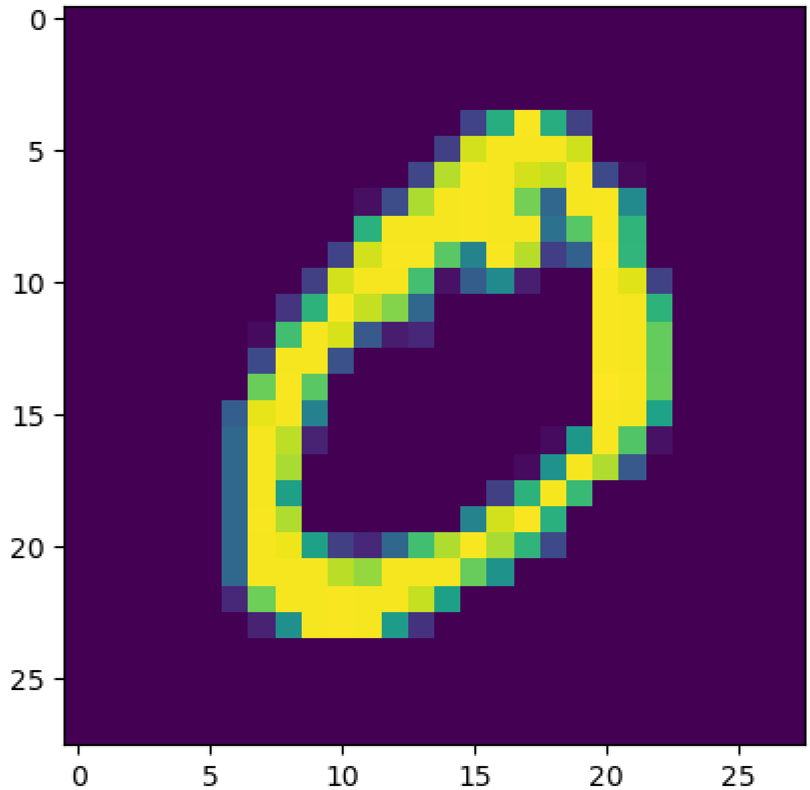
In [6]:
```
# New Image vs Old
# New Image with noise added
plt.imshow(image_noise_added.reshape(28, 28))
```

Out[6]: <matplotlib.image.AxesImage at 0x267d288a9e0>



In [7]:
```
# Old Image without noise added
plt.imshow(x_train[1].reshape(28, 28))
```

Out[7]: <matplotlib.image.AxesImage at 0x2678d3ebbe0>



In [8]:
```
# Normalization
mean = np.mean(x_train)
std = np.std(x_train)
x_train_norm = (x_train - mean) / std

mean = np.mean(x_test)
std = np.std(x_test)
x_test_norm = (x_test - mean) / std
```

In [9]:
```
### Add noise to whole set of images
# Noise is added here

# Generate random noise with the same shape as the image
import numpy as np
mean = .0
stddev = 2.0
# noise = np.random.normal(mean, stddev, x_train.shape)

# Generate random noise with the same shape as the image
# test_noise = np.random.normal(mean, stddev, x_test.shape)
x_train_noise = np.random.normal(mean, stddev, x_train.shape)
x_test_noise = np.random.normal(mean, stddev, x_test.shape)

# Add noise
x_train_noise_added = x_train_norm + x_train_noise
x_test_noise_added = x_test_norm + x_test_noise
# y_train_noise = y_train + y_train_noise
# y_test_noise = y_test + y_test_noise

# Add noise
x_test_noise[1]
```

Out[9]:
```
array([ 1.98526283e+00,  3.72791427e+00,  1.74883941e-01,  7.87424236e-01,
       -3.42958331e+00,  2.11577901e+00, -2.03260579e+00, -3.25397698e-01,
       -8.70449259e-01, -2.30001116e+00, -2.75004799e+00, -5.81128565e-01,
       -9.93027120e-01,  1.93561432e+00,  3.99198015e+00, -1.02125915e+00,
        9.74723778e-01,  1.14586365e+00,  4.67077591e-01,  2.28481688e+00,
       -2.33466069e+00, -2.17919131e+00, -3.75952181e+00, -1.01123996e+00,
       -1.37726083e+00,  2.79304529e+00, -1.20813826e+00, -5.05302952e+00,
        1.32992202e-01, -1.22959905e+00,  2.91419916e+00, -4.37376880e+00,
        1.98512085e+00,  6.29047567e-01, -2.96553610e+00,  1.75077724e+00,
       -3.95428718e+00, -6.01371711e-01, -2.91109500e+00,  4.70922606e-01,
       -3.94998918e+00, -1.20390627e+00, -5.44763946e-01, -4.28761970e-01,
        4.05453587e+00,  2.65797699e+00, -4.24462282e-01,  6.67972819e-01,
        1.90181279e+00, -4.07674993e-01, -1.23273354e+00,  1.47100583e+00,
        2.52791927e+00, -5.40026988e-01, -3.47834552e+00, -4.73983355e+00,
       -8.75565728e-01, -2.21249861e+00,  4.57310126e-02, -5.40797281e-01,
        1.39128235e+00, -9.46246816e-01,  3.76294070e+00,  1.28581168e+00,
        2.92603488e+00,  1.34538323e+00,  2.02639626e+00, -3.35336482e+00,
       -4.47495641e-02,  2.94689061e+00,  2.92764428e+00,  6.31827053e-01,
       -1.01202053e+00, -4.17295788e-01, -2.75129331e+00, -2.14951766e+00,
       -1.92343888e-01, -2.37683051e+00,  8.30792112e-01, -2.07047581e+00,
        6.49557336e-01,  1.29457458e+00, -1.10480834e+00, -1.42006802e+00,
        2.69639775e+00, -9.68997175e-01, -6.83588558e-01, -1.01902847e-01,
        2.47167743e+00,  1.82709885e+00,  2.65813353e-02,  6.83638552e-01,
        2.38015415e+00, -9.15276835e-01, -1.40067057e+00,  1.50022383e+00,
        5.90639640e-01,  7.26488739e-01,  3.47725292e+00,  2.92374034e-02,
        4.13613251e+00,  1.35965399e-01,  4.03602213e+00, -5.56841673e-01,
        1.95031646e+00, -3.35699218e+00, -3.52003406e+00, -1.67808347e+00,
        2.85468741e+00,  2.04736126e+00, -1.23998059e+00,  1.44938424e+00,
       -3.92744106e+00, -2.91379215e+00,  4.58355741e-01, -6.96909449e-01,
        1.28030213e+00,  1.77786793e+00, -8.76616765e-01, -2.53567284e+00,
        4.00256048e-01,  8.65873266e-01,  1.22034455e-01,  4.96999250e-01,
        5.64251511e-01,  1.93749835e-01, -1.49113591e+00, -3.63922841e+00,
       -5.08894848e-01,  2.87228302e+00, -2.09540469e+00,  2.69488793e+00,
       -1.64810625e+00, -2.83061715e-01,  4.18530918e+00, -1.91495944e-01,
        2.18386021e+00,  8.68488416e-01,  4.33981544e+00,  3.50369118e+00,
        3.20311612e+00,  4.64213506e-01, -5.44963151e-01, -2.93176259e+00,
       -1.74234738e+00,  1.54307814e+00, -3.54124718e-01, -2.65740817e+00,
       -2.12753899e+00, -7.99904141e-01,  2.48000559e+00, -1.43251544e+00,
        1.81450421e+00,  1.00107587e+00, -6.90534982e-01, -9.76334846e-01,
        3.89453456e-01, -5.86060436e+00,  3.44995405e+00, -1.23941682e+00,
       -8.41667452e-01, -1.61150915e+00, -5.77489323e+00, -2.15247860e+00,
        1.45838575e+00,  2.54242205e+00,  2.08876089e+00,  4.67493668e-01,
        3.69076428e-01, -3.39950294e+00,  2.82184799e+00, -6.95374261e-01,
        2.17230040e-01, -7.45236122e-01,  4.52665286e+00, -1.07516955e-01,
        1.04088329e+00,  1.44275793e-02, -6.57721030e-01, -2.65796030e+00,
        3.35305954e+00, -1.33938758e+00,  6.29696194e-01, -2.58356120e+00,
       -2.42989686e+00, -1.75223201e+00, -1.59430872e+00,  1.54319276e+00,
       -1.90232932e+00, -6.52365401e-01, -4.93543740e-02, -9.03254834e-01,
        1.32861174e+00,  7.97904831e-01,  1.72840306e+00, -4.25972606e-01,
        8.10241854e-01, -2.16838565e+00,  8.70154637e-01, -4.06059305e+00,
        3.33030505e+00,  1.20794994e+00, -1.38204300e+00, -6.89273846e-01,
        3.62342191e-01,  1.39599140e+00, -7.74614808e-01,  3.81792646e-01,
        5.62564749e-01, -3.56055404e+00,  8.30787106e-01, -1.15445497e+00,
       -4.80048954e-01,  1.83674509e+00, -5.78940783e-01,  6.81256548e-01,
       -3.41426069e+00, -2.68402268e+00, -3.95132393e+00,  1.00788225e+00,
        3.47676039e+00,  1.76620120e+00, -3.70728440e+00,  3.72114767e+00,
        6.77615830e-01,  9.47261687e-01, -1.32349435e+00, -1.81393303e+00,
       -2.25563533e+00, -1.00886927e+00,  1.97810282e+00,  2.15058186e-01,
        1.05654215e+00,  6.52281069e-01,  9.08715486e-01, -9.71217010e-01,
        7.72641386e-02, -6.32143098e-02, -1.47350719e+00,  4.25997726e-03,
```

```
 3.68254894e-01, -2.00731205e+00, -5.72958307e-01, -1.35093760e-01,
 1.07141798e+00, -1.70799194e-01,  8.25328428e-01, -3.71933931e+00,
-1.27863509e+00,  1.02902806e-01,  2.42597385e+00,  7.77998653e-01,
-1.70842311e+00,  9.92004808e-01, -1.88351673e+00, -1.82397394e+00,
-1.47743772e+00, -3.37385637e+00,  1.39276027e+00, -2.80554326e+00,
-2.09115768e+00,  8.50372513e-01,  4.54975265e+00, -3.90367214e-01,
 1.95490318e-01,  1.25031899e-01,  1.96595107e+00,  9.93791706e-01,
 2.80440433e+00,  8.82830839e-01,  7.04997738e-01,  9.21632387e-01,
-1.30773953e+00, -3.22437107e+00, -1.50616689e+00,  9.96547293e-01,
 3.45099148e+00, -1.48832545e+00,  1.08295803e+00,  1.85597277e+00,
 1.54664200e+00,  1.95470791e+00, -3.98820920e+00, -1.93480962e-04,
-1.86595632e-02, -6.31913602e-01, -9.78430521e-01, -1.74865583e+00,
 2.37136297e-01, -1.58238279e+00,  1.41512689e+00,  1.63322615e+00,
-1.71696836e+00,  2.26888384e+00, -3.22836686e+00, -2.61392399e+00,
-2.51285491e+00, -2.02847037e+00,  2.08346533e+00, -1.38044172e+00,
-2.57317153e+00,  1.15212652e+00, -3.18196807e+00,  3.90117818e+00,
 1.32030239e+00,  1.40024330e+00, -1.89826198e+00, -1.69256370e+00,
-2.25205279e-01, -7.81808069e-01,  1.84655040e-01,  3.76646608e+00,
 2.54749927e+00, -6.44866337e-01, -1.69101582e+00,  1.92613197e+00,
 1.44442616e+00,  1.41504741e+00, -9.34794752e-01,  7.31254444e-01,
-1.57718887e+00,  1.76425234e+00,  2.96614803e+00,  1.79871379e+00,
-4.36448917e-01, -4.18141408e+00, -4.15473743e+00,  4.36121311e-01,
-7.27525365e-01,  4.14523626e-02,  1.43484511e-01,  1.24973716e+00,
-8.52192636e-01, -1.32354249e+00, -5.97091513e-01,  3.58804186e+00,
 2.35348017e+00,  2.07103654e-01,  1.65617297e+00, -4.78850098e-01,
-8.26440117e-01,  1.19220348e+00, -1.20323986e+00, -8.41028174e-01,
-1.70757984e+00,  7.63804018e-02,  1.18249586e+00,  1.81293837e+00,
 1.26790009e+00,  3.43254569e+00, -4.18498504e-01,  8.73025354e-01,
-4.40236801e-01, -3.82642357e-02, -7.20322948e-01, -8.52574536e-01,
-5.50759227e+00, -1.59517175e+00,  1.89974319e-01, -5.58899945e-01,
 9.95986320e-01, -1.31261796e+00,  3.01940573e+00,  2.58517504e+00,
 3.73859403e+00, -2.62397236e+00, -1.47163498e-01,  1.25763644e+00,
-2.07955908e-01,  8.92460802e-01, -1.16898951e+00,  2.93324297e+00,
 3.94403997e-01,  1.04131627e+00, -1.03254731e+00,  2.32561924e+00,
-2.25625690e+00, -1.16711442e+00,  1.49706815e+00,  4.26071886e-01,
 1.01985405e+00, -2.67794680e+00, -8.53711599e-01, -1.18773539e+00,
 9.42771008e-01, -2.99879757e+00,  9.22720569e-01, -5.19382669e+00,
-3.05771946e+00,  1.85015919e-01,  4.05185086e-01,  1.31182714e+00,
-2.65934588e+00,  3.47184628e+00, -8.78072037e-01, -3.53609035e-01,
-3.28395800e+00,  2.24896527e+00,  5.31343743e-01, -1.28513638e-01,
 1.64439131e+00, -1.87763415e+00, -2.75177855e+00, -1.94896226e+00,
-1.77279657e+00,  3.73962968e-01,  1.40179681e+00,  9.57689309e-01,
 1.89641163e+00,  2.11737706e+00, -1.81441217e+00, -1.50217009e+00,
 1.74362290e-01, -1.03720873e+00, -1.40012923e+00, -4.05973486e-01,
 2.22283990e-01, -2.00135199e+00, -4.53656229e-01,  2.17107506e+00,
 2.79231366e+00, -2.37635852e-01,  1.91916090e+00,  1.85390477e+00,
-2.09039603e+00,  2.85999390e-01, -1.65235688e+00, -5.40237159e-01,
-5.10943753e-01,  4.39272968e+00, -5.39299105e-01,  3.56613848e+00,
-2.20662614e+00, -5.45957661e-01,  7.26330331e-01, -1.73254073e-01,
-2.36524365e-01, -2.05668065e+00,  5.91654671e-01, -7.26742122e-01,
 3.76130257e-01, -3.78156880e+00, -1.27261757e+00,  1.56512368e-01,
-2.66035883e-01,  1.13364976e-01,  2.55012924e+00, -1.25979685e+00,
 2.07090834e+00, -3.31725814e-01,  1.68690764e-01, -2.53719369e+00,
 2.22598226e+00, -1.43089533e-01, -6.62971762e+00,  7.52711514e-01,
 3.10526621e+00,  4.09887088e+00,  2.53048762e-01, -2.54059536e+00,
-9.85483798e-01, -1.58768878e+00, -4.07260184e-01,  5.29298689e-01,
 3.26027074e+00,  9.83136155e-01, -7.62310737e-01, -2.81678471e+00,
 2.54486196e+00, -1.32068943e+00, -4.27978712e-01, -1.86953491e+00,
 2.66018675e+00, -1.28662322e+00,  3.86484021e+00,  7.54365487e-02,
 3.02234421e-01,  2.86463311e+00, -1.35867610e+00,  1.55690498e+00,
```

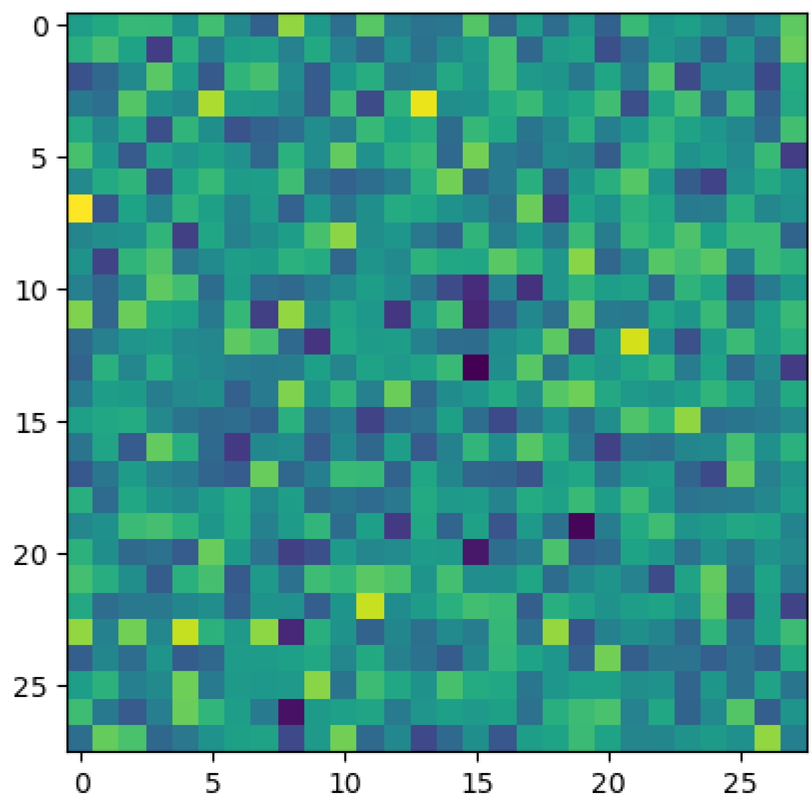
```
 3.31003733e+00,  1.03984635e+00,  2.11786941e+00, -1.93758206e+00,
-1.77365728e+00,  2.06481442e+00, -9.87892244e-01,  3.74985721e+00,
 4.24477162e-01,  1.27725768e+00,  5.05718024e-01, -1.44600402e+00,
-4.11934759e-01, -3.10140891e-01,  2.25352514e+00,  6.00208928e-01,
-7.59265335e-01,  1.61376959e+00,  2.53903311e-01,  3.03805243e+00,
-3.55808694e+00,  7.99527586e-01, -1.95887907e+00, -2.77067121e+00,
 2.14324774e-01,  3.50258510e-01,  5.43189321e+00, -5.05143652e-01,
 8.72342776e-01,  4.32539550e+00,  4.88565213e+00,  4.06111502e+00,
-1.74100715e+00, -3.59139969e-01,  2.50582041e+00, -6.57894551e-01,
-3.60048103e+00, -7.62943286e-01, -8.18747860e-01,  2.57168568e+00,
-2.44871193e+00,  6.64945434e-01,  3.23211451e+00,  1.72011548e-01,
-2.52374519e+00, -1.53610505e-01,  6.40575313e-01, -1.69757483e+00,
-9.94328624e-01,  1.80641360e+00, -6.16913945e-01, -1.66678390e+00,
-5.58182361e-01,  5.49238045e-01,  2.40453854e+00,  8.19856661e-01,
-1.44637253e+00, -2.66370421e+00,  1.47401859e-01,  2.91299553e+00,
-1.97884558e+00, -1.49130651e+00,  1.07615953e+00, -1.93889306e+00,
 9.76109523e-01,  8.25209250e-01, -1.61559292e+00, -1.13250749e+00,
-5.07465118e+00,  1.89476178e+00, -1.92574309e+00,  1.52701246e-01,
-6.84202559e-01,  3.29379351e-01,  2.79898860e+00, -5.54152717e-01,
 1.05331115e+00, -9.18079609e-01, -1.07329241e+00,  1.18062419e+00,
 1.04598862e+00, -1.32614540e-01, -3.00900508e+00,  4.01805876e+00,
-1.47373605e+00, -1.89034895e+00, -3.02431968e+00, -1.61332126e+00,
-2.16571178e+00,  3.85511239e+00, -2.69608763e+00,  3.18758784e-01,
-2.16913000e+00, -1.47368378e+00, -9.19880657e-01, -6.48784840e-01,
-3.05054044e+00,  1.02054496e+00,  4.43698287e-01, -1.23684418e+00,
-6.06301788e-01,  2.73879102e+00, -3.07327364e+00, -3.24864500e-01,
-2.59692994e+00, -2.23836469e+00,  2.98572656e-01,  2.24000385e+00,
 2.53399294e+00, -1.26067958e+00, -5.43354903e-01, -3.65317872e+00,
 7.34430762e-01, -2.77978054e-01,  8.79550060e-01,  1.69717146e+00,
-1.64404431e-01, -3.88933884e-01,  5.39199023e-01,  1.69196345e+00,
 3.99273972e-01,  1.10256874e-01,  8.61632821e-01,  2.54139483e+00,
 1.27206794e+00, -2.20010823e+00, -5.75957438e-01, -1.08775140e+00,
 2.01907861e-01,  2.60304239e+00, -1.09504881e+00, -4.19545727e+00,
-2.14063692e+00, -1.51984285e+00, -1.12975685e+00, -4.72569610e+00,
 2.71447731e-02,  3.47233107e+00, -2.02253282e+00, -7.91935671e-01,
-5.15188396e-01,  8.82611765e-01,  1.76599586e-01,  7.09898168e-01,
-2.09644093e+00,  3.46186370e+00,  9.35571822e-01, -1.26309554e+00,
-3.66147953e+00,  1.39877069e+00,  1.70838292e+00,  1.20858910e+00,
 2.56265212e+00, -1.08278248e+00,  1.47862849e+00,  1.44852959e+00,
 2.47260264e+00, -5.13007929e-01, -1.62615444e-01,  1.28952990e+00,
-4.22666432e+00,  2.01302922e+00,  2.77750760e+00, -3.23531097e+00,
-4.97271464e+00, -3.67050280e-01, -1.34134816e+00,  9.59558649e-01,
-6.12018273e-01, -4.80938489e-01, -3.57087546e+00, -2.76209600e+00,
 5.02872179e-02,  1.13830270e-01, -2.61090499e+00,  3.23256000e+00,
 4.03603298e+00, -1.94963850e+00,  2.70997673e+00, -2.95580273e-01,
 8.77840966e-01,  1.78284126e+00, -4.16936090e+00, -1.87141672e+00,
-1.24722444e+00,  2.07211240e+00, -4.44212381e+00,  1.40826211e+00,
-2.17718867e+00, -3.46292437e-02,  1.45459799e+00, -1.10595510e+00,
-1.21584142e+00, -5.77601508e-01, -1.15135249e+00, -3.70065738e-01,
 7.40347092e-01,  3.35329450e+00, -4.87795480e-01,  8.00202268e-01,
-2.31027615e+00,  4.34550368e+00,  3.69414086e+00,  3.89241130e+00,
-2.68274851e+00,  4.37192470e+00,  1.76901968e+00,  2.28860584e+00,
-3.42288779e+00,  7.87254274e-01,  6.50906456e-01,  5.15458210e-01,
-4.08233522e-01,  1.42229094e-02,  1.18868263e+00,  6.35880274e-01,
-1.69164097e+00, -5.26696606e-01,  1.79027910e+00,  1.51925002e+00,
-1.37507793e+00,  3.00375281e+00,  5.73601186e-01, -3.40245156e+00,
-1.44041484e+00,  1.20551566e+00, -1.63067972e+00,  4.14710747e-01,
-5.94337788e-01,  1.24214939e+00, -1.47142467e+00, -4.79015548e-01,
-8.49983629e-01, -1.69511851e+00, -9.44350614e-01,  7.81777747e-01,
-6.47976679e-01, -4.94554077e+00,  1.25869319e+00, -2.57331946e+00,
```

```
 2.91115501e+00,  1.53627230e+00,  7.95663228e-01, -6.49879035e-01,
 2.47322911e+00,  5.78016164e-01, -9.35640854e-01,  7.06282749e-01,
 7.59100656e-01, -2.08039361e+00,  1.07705670e+00, -3.15859978e+00,
 2.36673955e+00,  2.13134044e+00,  8.01320448e-01, -4.11869548e-01,
-4.29252739e-01,  3.79287819e+00,  2.68601173e+00, -1.71729191e-02,
 8.74836584e-01, -6.90105117e-01, -3.73782991e-01,  1.15029076e-01,
 1.76076156e+00, -2.40512317e+00, -8.91663751e-01, -3.88712580e+00,
 2.61109615e+00, -1.71419144e+00, -1.52867955e+00, -3.06506129e+00,
-2.01102502e-01,  5.61763729e-01,  5.06148012e+00, -2.49460669e+00,
-6.48952380e-01, -6.14372085e-01, -9.96071107e-01, -3.32742161e+00,
 2.72131263e+00, -3.97473527e+00, -6.32709553e-01, -7.73707871e-01,
-7.54186900e-01,  3.48942809e-01, -2.48242842e-01, -1.83642218e-01,
 3.82953379e-01, -2.66003652e+00,  4.49717641e-01,  3.07856272e-01,
 2.36621262e+00, -3.23380139e-01,  5.83108060e-01,  2.95828334e+00,
-2.82613732e+00, -1.03981455e+00, -3.36464120e-01,  1.84188986e+00,
 6.90461758e-01, -2.19475851e+00,  8.27730946e-01, -6.12220266e-01])
```

In [10]:

```
# Compare Noises to Ensure they are Different
# Noise on Image 122
plt.imshow(x_train_noise[122].reshape(28, 28))
```

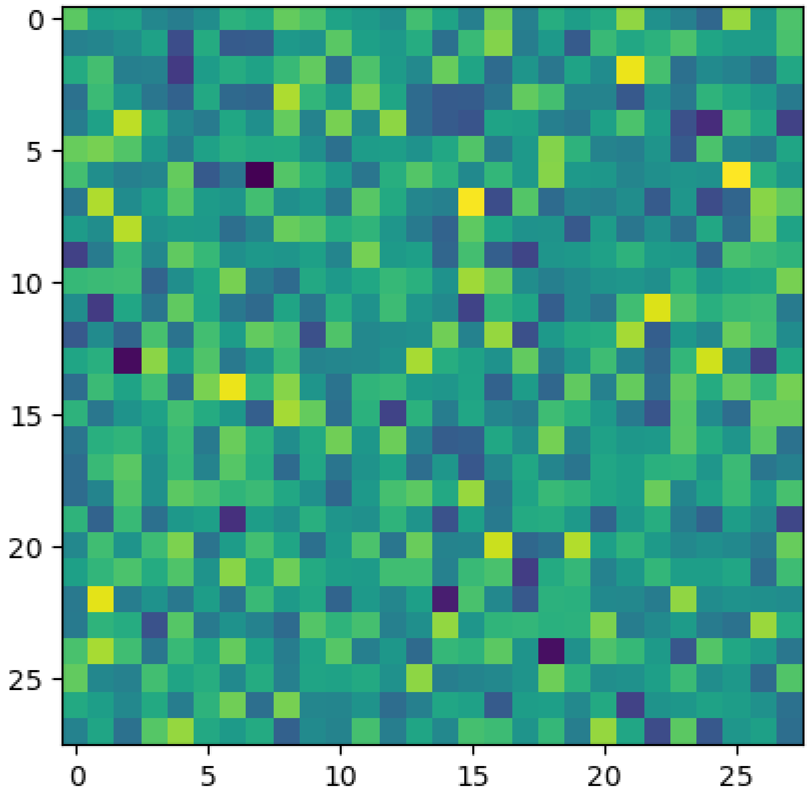
Out[10]: <matplotlib.image.AxesImage at 0x2678d459b10>



In [11]:

```
# Noise on Image 124
plt.imshow(x_train_noise[124].reshape(28, 28))
```

Out[11]: <matplotlib.image.AxesImage at 0x2678d4de380>



As the two visual representations (images) of the noise show different patterns, they are indicating different noise levels for each image

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

In [12]:
```
%%capture
## Code From Base for Creating Neural Network Without Noise
def neural_network(batch_size, num_classes, epochs, x_train, x_test, y_train, y_test):

    # 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="adam",
                  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 score
```

In [13]:

```
import pandas as pd
(
np.max(x_train_noise),
np.min(x_train_noise),
np.max(x_train),
np.min(x_train)
)
```

Out[13]: (10.62403342399543, -11.502094450839351, 1.0, 0.0)

In [14]:

```
batch_size = 128
num_classes = 10
epochs = 20

# NN With Noise
neural_network(batch_size, num_classes, epochs, x_train_noise_added, x_test_noise_adde
# NN Without Noise
neural_network(batch_size, num_classes, epochs, x_train, x_test, y_train, y_test)
```

```
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 6s 11ms/step - loss: 0.7483 - accuracy: 0.
7568 - val_loss: 0.4324 - val_accuracy: 0.8569
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3418 - accuracy: 0.
8888 - val_loss: 0.3739 - val_accuracy: 0.8772
Epoch 3/20
469/469 [==============================] - 5s 11ms/step - loss: 0.2089 - accuracy: 0.
9293 - val_loss: 0.3616 - val_accuracy: 0.8841
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1315 - accuracy: 0.
9553 - val_loss: 0.4020 - val_accuracy: 0.8861
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1046 - accuracy: 0.
9639 - val_loss: 0.4299 - val_accuracy: 0.8849
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0807 - accuracy: 0.
9722 - val_loss: 0.4744 - val_accuracy: 0.8830
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0756 - accuracy: 0.
9737 - val_loss: 0.4915 - val_accuracy: 0.8847
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0683 - accuracy: 0.
9769 - val_loss: 0.5140 - val_accuracy: 0.8910
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0632 - accuracy: 0.
9788 - val_loss: 0.5464 - val_accuracy: 0.8889
Epoch 10/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0580 - accuracy: 0.
9808 - val_loss: 0.5866 - val_accuracy: 0.8845
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0535 - accuracy: 0.
9815 - val_loss: 0.6281 - val_accuracy: 0.8823
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0530 - accuracy: 0.
9828 - val_loss: 0.6033 - val_accuracy: 0.8864
Epoch 13/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0511 - accuracy: 0.
9832 - val_loss: 0.6275 - val_accuracy: 0.8866
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0539 - accuracy: 0.
```

```
9829 - val_loss: 0.6465 - val_accuracy: 0.8913
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0473 - accuracy: 0.
9847 - val_loss: 0.6660 - val_accuracy: 0.8896
Epoch 16/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0471 - accuracy: 0.
9853 - val_loss: 0.6574 - val_accuracy: 0.8917
Epoch 17/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0488 - accuracy: 0.
9847 - val_loss: 0.6506 - val_accuracy: 0.8936
Epoch 18/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0457 - accuracy: 0.
9863 - val_loss: 0.6902 - val_accuracy: 0.8907
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0408 - accuracy: 0.
9879 - val_loss: 0.7065 - val_accuracy: 0.8918
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0424 - accuracy: 0.
9871 - val_loss: 0.7421 - val_accuracy: 0.8897
Test loss: 0.7421275973320007
Test accuracy: 0.8896999955177307
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2468 - accuracy: 0.
9263 - val_loss: 0.1069 - val_accuracy: 0.9669
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1028 - accuracy: 0.
9685 - val_loss: 0.0781 - val_accuracy: 0.9759
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0699 - accuracy: 0.
9785 - val_loss: 0.0716 - val_accuracy: 0.9782
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0557 - accuracy: 0.
9821 - val_loss: 0.0731 - val_accuracy: 0.9780
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0453 - accuracy: 0.
9857 - val_loss: 0.0613 - val_accuracy: 0.9818
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0388 - accuracy: 0.
9871 - val_loss: 0.0709 - val_accuracy: 0.9794
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0355 - accuracy: 0.
```

```
9883 - val_loss: 0.0731 - val_accuracy: 0.9790
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0304 - accuracy: 0.
9903 - val_loss: 0.0612 - val_accuracy: 0.9831
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0267 - accuracy: 0.
9913 - val_loss: 0.0659 - val_accuracy: 0.9820
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0265 - accuracy: 0.
9912 - val_loss: 0.0771 - val_accuracy: 0.9797
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0230 - accuracy: 0.
9920 - val_loss: 0.0812 - val_accuracy: 0.9815
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0209 - accuracy: 0.
9928 - val_loss: 0.0775 - val_accuracy: 0.9828
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0231 - accuracy: 0.
9926 - val_loss: 0.0726 - val_accuracy: 0.9820
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0188 - accuracy: 0.
9934 - val_loss: 0.0784 - val_accuracy: 0.9811
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0198 - accuracy: 0.
9932 - val_loss: 0.0831 - val_accuracy: 0.9820
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0203 - accuracy: 0.
9931 - val_loss: 0.0812 - val_accuracy: 0.9817
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0144 - accuracy: 0.
9952 - val_loss: 0.0703 - val_accuracy: 0.9838
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0161 - accuracy: 0.
9947 - val_loss: 0.0785 - val_accuracy: 0.9838
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0166 - accuracy: 0.
9944 - val_loss: 0.0825 - val_accuracy: 0.9833
Epoch 20/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0153 - accuracy: 0.
9949 - val_loss: 0.0730 - val_accuracy: 0.9852
Test loss: 0.0729510709643364
Test accuracy: 0.9851999878883362
```

Out[14]:
```
[0.0729510709643364, 0.9851999878883362]
```

Looking at the above two models generated by neural nets, we see that the model with noise reached a final accuracy of .889, while the one without noise reached a final accuracy of .985. All other elements of the model, besides noise, are held constant (the function defines the same number of layers, and batch size/number of classes/epochs are also constant.) Therefore, we see a noticeable dropoff in accuracy as a result of adding noise to the dataset.

1. 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 [15]:
```
def neural_network_with_noise(mean, stddev, batch_size, num_classes, epochs, x_train,
    # noise = np.random.normal(mean, stddev, x_train.shape)

    # Generate random noise with the same shape as the image
    # test_noise = np.random.normal(mean, stddev, x_test.shape)
    x_train_noise = np.random.normal(mean, stddev, x_train.shape)
    x_test_noise = np.random.normal(mean, stddev, x_test.shape)

    # Add noise
    x_train_noise_added = x_train_norm + x_train_noise
    x_test_noise_added = x_test_norm + x_test_noise
    score = neural_network(batch_size, num_classes, epochs, x_train_noise_added, x_tes

    # Append Loss & Accuracy To List
    test_loss = score[0]
    test_accuracy = score[1]
    loss_values.append(test_loss)
    accuracy_values.append(test_accuracy)
```

In [16]:
```
# %%capture
## Combining Noise Code with Model-Running code for a single, repeatable block:
# Listing Scale (i.e. Standard Deviation) Values over which to train models
stddev_values = [0.1, 0.5, 1.0, 2.0, 4.0]  # Values given
# Initializing Lists
loss_values = []
accuracy_values = []
batch_size = 128
num_classes = 10
epochs = 20
mean = 0

# Loop:
for stddev in stddev_values:
    neural_network_with_noise(mean, stddev, batch_size, num_classes, epochs, x_train,
```

```
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 6s 12ms/step - loss: 0.2415 - accuracy: 0.
9262 - val_loss: 0.1186 - val_accuracy: 0.9622
Epoch 2/20
469/469 [==============================] - 5s 11ms/step - loss: 0.1122 - accuracy: 0.
9660 - val_loss: 0.0886 - val_accuracy: 0.9717
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0796 - accuracy: 0.
9749 - val_loss: 0.0802 - val_accuracy: 0.9761
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0617 - accuracy: 0.
9798 - val_loss: 0.0711 - val_accuracy: 0.9793
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0551 - accuracy: 0.
9818 - val_loss: 0.0683 - val_accuracy: 0.9799
Epoch 6/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0455 - accuracy: 0.
9852 - val_loss: 0.0767 - val_accuracy: 0.9780
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0430 - accuracy: 0.
9858 - val_loss: 0.0727 - val_accuracy: 0.9780
Epoch 8/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0386 - accuracy: 0.
9874 - val_loss: 0.0802 - val_accuracy: 0.9794
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0385 - accuracy: 0.
9872 - val_loss: 0.0801 - val_accuracy: 0.9790
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0356 - accuracy: 0.
9879 - val_loss: 0.0717 - val_accuracy: 0.9814
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0317 - accuracy: 0.
9896 - val_loss: 0.0704 - val_accuracy: 0.9818
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0280 - accuracy: 0.
9905 - val_loss: 0.0825 - val_accuracy: 0.9791
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0302 - accuracy: 0.
9902 - val_loss: 0.0913 - val_accuracy: 0.9803
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0315 - accuracy: 0.
```

```
9904 - val_loss: 0.0758 - val_accuracy: 0.9824
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0290 - accuracy: 0.
9911 - val_loss: 0.0851 - val_accuracy: 0.9829
Epoch 16/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0249 - accuracy: 0.
9920 - val_loss: 0.0837 - val_accuracy: 0.9819
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0242 - accuracy: 0.
9924 - val_loss: 0.0807 - val_accuracy: 0.9819
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0252 - accuracy: 0.
9923 - val_loss: 0.1018 - val_accuracy: 0.9799
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0240 - accuracy: 0.
9926 - val_loss: 0.0854 - val_accuracy: 0.9834
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0195 - accuracy: 0.
9940 - val_loss: 0.0910 - val_accuracy: 0.9823
Test loss: 0.09101765602827072
Test accuracy: 0.9822999835014343
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 6s 11ms/step - loss: 0.2759 - accuracy: 0.
9122 - val_loss: 0.1246 - val_accuracy: 0.9610
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1058 - accuracy: 0.
9664 - val_loss: 0.1049 - val_accuracy: 0.9679
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0705 - accuracy: 0.
9772 - val_loss: 0.0926 - val_accuracy: 0.9720
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0476 - accuracy: 0.
9849 - val_loss: 0.1047 - val_accuracy: 0.9696
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0401 - accuracy: 0.
9865 - val_loss: 0.1101 - val_accuracy: 0.9687
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0388 - accuracy: 0.
9870 - val_loss: 0.1136 - val_accuracy: 0.9689
Epoch 7/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0308 - accuracy: 0.
```

```
9902 - val_loss: 0.1009 - val_accuracy: 0.9735
Epoch 8/20
469/469 [==============================] - 6s 12ms/step - loss: 0.0260 - accuracy: 0.
9915 - val_loss: 0.1080 - val_accuracy: 0.9743
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0262 - accuracy: 0.
9917 - val_loss: 0.1426 - val_accuracy: 0.9680
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0255 - accuracy: 0.
9919 - val_loss: 0.1264 - val_accuracy: 0.9726
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0215 - accuracy: 0.
9930 - val_loss: 0.1437 - val_accuracy: 0.9698
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0274 - accuracy: 0.
9917 - val_loss: 0.1331 - val_accuracy: 0.9720
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0185 - accuracy: 0.
9941 - val_loss: 0.1346 - val_accuracy: 0.9737
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0234 - accuracy: 0.
9926 - val_loss: 0.1297 - val_accuracy: 0.9743
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0210 - accuracy: 0.
9938 - val_loss: 0.1399 - val_accuracy: 0.9729
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0207 - accuracy: 0.
9937 - val_loss: 0.1222 - val_accuracy: 0.9763
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0182 - accuracy: 0.
9948 - val_loss: 0.1423 - val_accuracy: 0.9749
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0197 - accuracy: 0.
9940 - val_loss: 0.1558 - val_accuracy: 0.9738
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0154 - accuracy: 0.
9956 - val_loss: 0.1599 - val_accuracy: 0.9743
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0171 - accuracy: 0.
9950 - val_loss: 0.1560 - val_accuracy: 0.9754
Test loss: 0.15600836277008057
Test accuracy: 0.9753999710083008
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
```

```
Epoch 1/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3891 - accuracy: 0.
8780 - val_loss: 0.1948 - val_accuracy: 0.9382
Epoch 2/20
469/469 [==============================] - 4s 10ms/step - loss: 0.1477 - accuracy: 0.
9541 - val_loss: 0.1660 - val_accuracy: 0.9468
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0817 - accuracy: 0.
9740 - val_loss: 0.1656 - val_accuracy: 0.9499
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0539 - accuracy: 0.
9822 - val_loss: 0.1733 - val_accuracy: 0.9531
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0416 - accuracy: 0.
9857 - val_loss: 0.1802 - val_accuracy: 0.9551
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0363 - accuracy: 0.
9880 - val_loss: 0.2054 - val_accuracy: 0.9519
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0375 - accuracy: 0.
9878 - val_loss: 0.2002 - val_accuracy: 0.9522
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0296 - accuracy: 0.
9898 - val_loss: 0.2361 - val_accuracy: 0.9533
Epoch 9/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0312 - accuracy: 0.
9897 - val_loss: 0.2237 - val_accuracy: 0.9536
Epoch 10/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0262 - accuracy: 0.
9911 - val_loss: 0.2388 - val_accuracy: 0.9549
Epoch 11/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0279 - accuracy: 0.
9914 - val_loss: 0.2580 - val_accuracy: 0.9525
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0255 - accuracy: 0.
9923 - val_loss: 0.2539 - val_accuracy: 0.9541
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0222 - accuracy: 0.
9929 - val_loss: 0.2700 - val_accuracy: 0.9537
Epoch 14/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0239 - accuracy: 0.
9926 - val_loss: 0.2607 - val_accuracy: 0.9555
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0249 - accuracy: 0.
9924 - val_loss: 0.2821 - val_accuracy: 0.9547
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0236 - accuracy: 0.
9932 - val_loss: 0.2635 - val_accuracy: 0.9561
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0191 - accuracy: 0.
9945 - val_loss: 0.2725 - val_accuracy: 0.9557
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0224 - accuracy: 0.
9930 - val_loss: 0.3139 - val_accuracy: 0.9499
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0206 - accuracy: 0.
9939 - val_loss: 0.2683 - val_accuracy: 0.9569
Epoch 20/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0191 - accuracy: 0.
```

```
9943 - val_loss: 0.2939 - val_accuracy: 0.9562
Test loss: 0.29390591382980347
Test accuracy: 0.9562000036239624
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 6s 12ms/step - loss: 0.7541 - accuracy: 0.
7565 - val_loss: 0.4416 - val_accuracy: 0.8556
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3428 - accuracy: 0.
8869 - val_loss: 0.3733 - val_accuracy: 0.8791
Epoch 3/20
469/469 [==============================] - 5s 11ms/step - loss: 0.2065 - accuracy: 0.
9302 - val_loss: 0.3729 - val_accuracy: 0.8886
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1373 - accuracy: 0.
9537 - val_loss: 0.3961 - val_accuracy: 0.8871
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1003 - accuracy: 0.
9656 - val_loss: 0.4312 - val_accuracy: 0.8845
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0822 - accuracy: 0.
9715 - val_loss: 0.4945 - val_accuracy: 0.8839
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0743 - accuracy: 0.
9744 - val_loss: 0.4691 - val_accuracy: 0.8870
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0638 - accuracy: 0.
9786 - val_loss: 0.5123 - val_accuracy: 0.8865
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0652 - accuracy: 0.
9792 - val_loss: 0.4948 - val_accuracy: 0.8902
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0597 - accuracy: 0.
9802 - val_loss: 0.5283 - val_accuracy: 0.8908
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0582 - accuracy: 0.
9805 - val_loss: 0.5619 - val_accuracy: 0.8915
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0525 - accuracy: 0.
9826 - val_loss: 0.5962 - val_accuracy: 0.8840
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0546 - accuracy: 0.
```

```
9825 - val_loss: 0.5892 - val_accuracy: 0.8900
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0464 - accuracy: 0.
9852 - val_loss: 0.5928 - val_accuracy: 0.8901
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0413 - accuracy: 0.
9865 - val_loss: 0.6508 - val_accuracy: 0.8928
Epoch 16/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0477 - accuracy: 0.
9854 - val_loss: 0.6707 - val_accuracy: 0.8869
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0539 - accuracy: 0.
9834 - val_loss: 0.6592 - val_accuracy: 0.8868
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0441 - accuracy: 0.
9861 - val_loss: 0.6673 - val_accuracy: 0.8913
Epoch 19/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0442 - accuracy: 0.
9862 - val_loss: 0.6828 - val_accuracy: 0.8872
Epoch 20/20
469/469 [==============================] - 5s 11ms/step - loss: 0.0420 - accuracy: 0.
9874 - val_loss: 0.6794 - val_accuracy: 0.8881
Test loss: 0.6794248819351196
Test accuracy: 0.8881000280380249
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: 669706 (2.55 MB)
Trainable params: 669706 (2.55 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 6s 11ms/step - loss: 1.5456 - accuracy: 0.
4954 - val_loss: 1.0872 - val_accuracy: 0.6321
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.9989 - accuracy: 0.
6605 - val_loss: 1.0231 - val_accuracy: 0.6496
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.8121 - accuracy: 0.
7224 - val_loss: 1.0059 - val_accuracy: 0.6597
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.6572 - accuracy: 0.
7731 - val_loss: 1.0209 - val_accuracy: 0.6565
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.5320 - accuracy: 0.
8143 - val_loss: 1.0954 - val_accuracy: 0.6571
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.4400 - accuracy: 0.
```

```
8466 - val_loss: 1.1370 - val_accuracy: 0.6536
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3689 - accuracy: 0.
8707 - val_loss: 1.2539 - val_accuracy: 0.6518
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3295 - accuracy: 0.
8855 - val_loss: 1.2811 - val_accuracy: 0.6530
Epoch 9/20
469/469 [==============================] - 5s 11ms/step - loss: 0.2873 - accuracy: 0.
9006 - val_loss: 1.3865 - val_accuracy: 0.6484
Epoch 10/20
469/469 [==============================] - 5s 12ms/step - loss: 0.2623 - accuracy: 0.
9101 - val_loss: 1.4220 - val_accuracy: 0.6516
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2420 - accuracy: 0.
9180 - val_loss: 1.4670 - val_accuracy: 0.6514
Epoch 12/20
469/469 [==============================] - 5s 11ms/step - loss: 0.2193 - accuracy: 0.
9252 - val_loss: 1.5364 - val_accuracy: 0.6553
Epoch 13/20
469/469 [==============================] - 5s 11ms/step - loss: 0.2126 - accuracy: 0.
9280 - val_loss: 1.5795 - val_accuracy: 0.6548
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2027 - accuracy: 0.
9316 - val_loss: 1.6368 - val_accuracy: 0.6538
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1904 - accuracy: 0.
9366 - val_loss: 1.6660 - val_accuracy: 0.6554
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1875 - accuracy: 0.
9379 - val_loss: 1.7188 - val_accuracy: 0.6529
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1777 - accuracy: 0.
9413 - val_loss: 1.7691 - val_accuracy: 0.6571
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1742 - accuracy: 0.
9427 - val_loss: 1.8267 - val_accuracy: 0.6528
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1665 - accuracy: 0.
9452 - val_loss: 1.7963 - val_accuracy: 0.6553
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1622 - accuracy: 0.
9477 - val_loss: 1.8360 - val_accuracy: 0.6524
Test loss: 1.8360189199447632
Test accuracy: 0.652400016784668
```

In [17]:

```
accuracy_values
```

Out[17]:

```
[0.9822999835014343,
 0.9753999710083008,
 0.9562000036239624,
 0.8881000280380249,
 0.652400016784668]
```

In [18]:

```
# Compile Loss and Accuracy statistics for the models into a single data frame
accuracy_loss_results_dataframe = pd.DataFrame({'stddev_values': stddev_values, 'loss_
accuracy_loss_results_dataframe
```

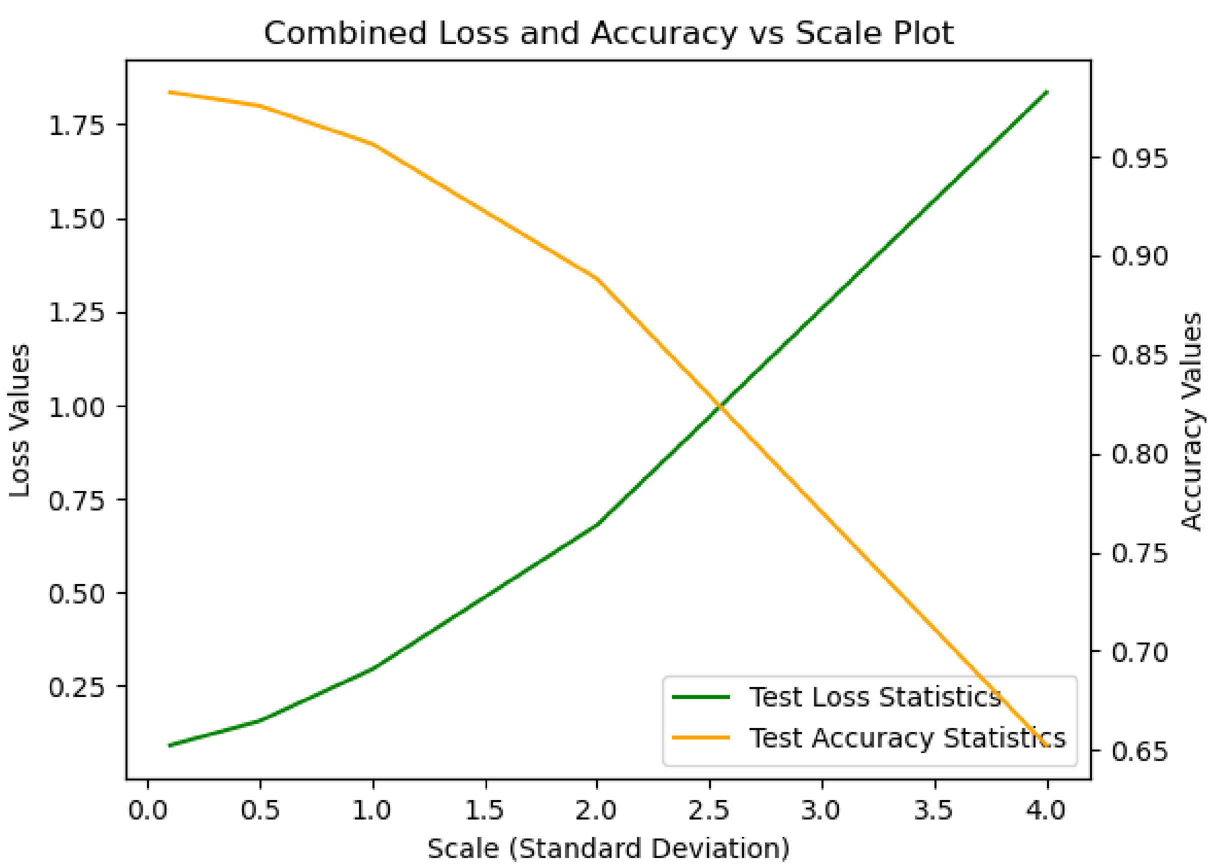
Out[18]:

	stddev_values	loss_values	accuracy_values
0	0.1	0.091018	0.9823
1	0.5	0.156008	0.9754
2	1.0	0.293906	0.9562
3	2.0	0.679425	0.8881
4	4.0	1.836019	0.6524

In [19]:

```
# Plotting Results
fig, ax1 = plt.subplots()
loss_line = ax1.plot(stddev_values, loss_values, color = 'green', label = 'Test Loss S

# Loss line Plot
ax1.set_xlabel('Scale (Standard Deviation)')
ax1.set_ylabel('Loss Values')
ax1.set_title('Combined Loss and Accuracy vs Scale Plot')

# Accuracy line Plot
ax2 = ax1.twinx()
accuracy_line = ax2.plot(stddev_values, accuracy_values, color='orange', label='Test A
ax2.set_ylabel('Accuracy Values')

# Combine Both Lines Into One Plot
lines = loss_line + accuracy_line
labels = [l.get_label() for l in lines]
ax1.legend(lines, labels, loc='lower right')
```

Out[19]: <matplotlib.legend.Legend at 0x2678d4c6950>



The plot shows an inverse relationship between the scale of the noise and the accuracy of the model; in other words, noise makes the models less predictive.

In [ ]:

```

```