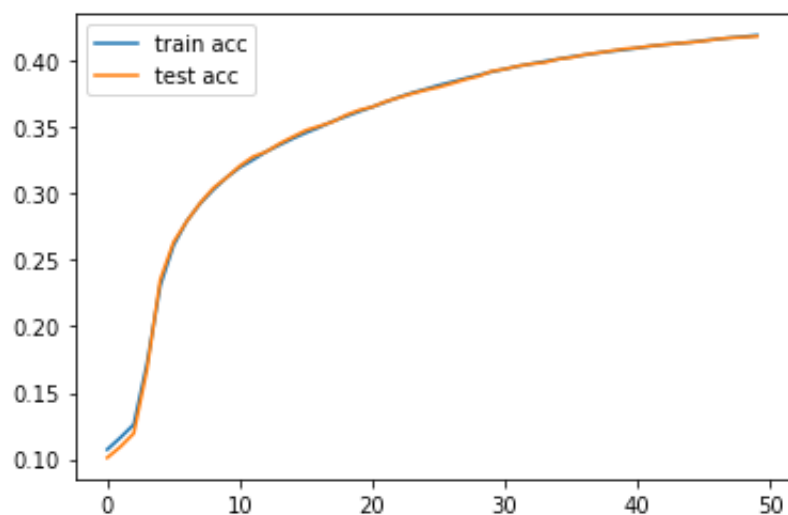
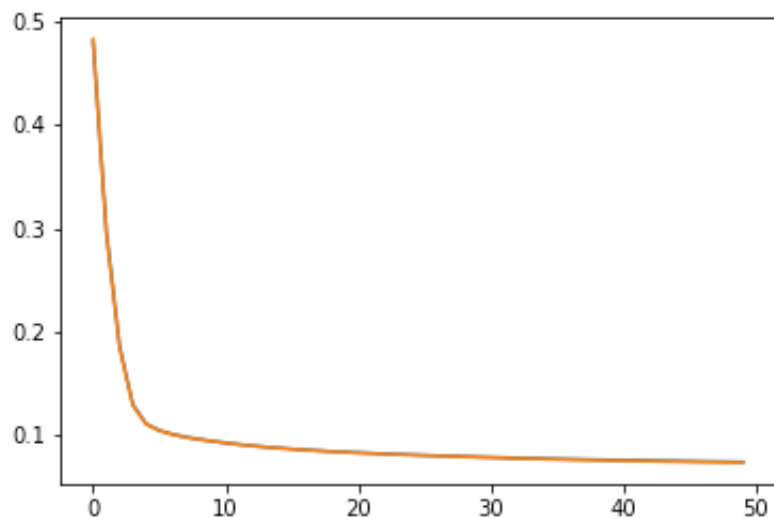
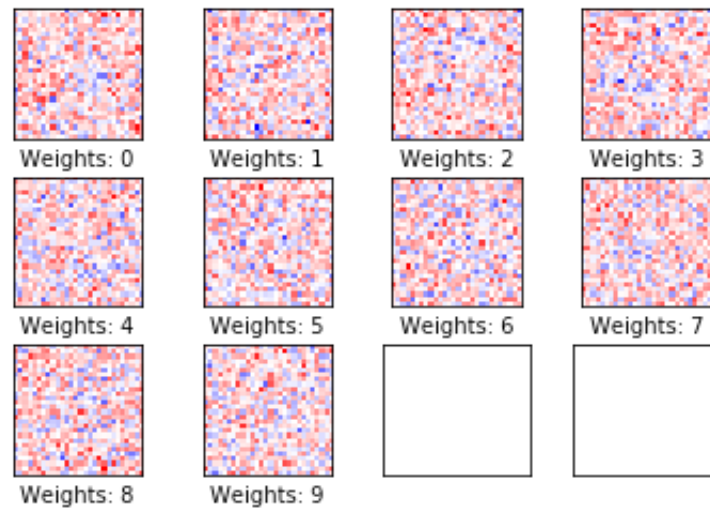


Logistische Regression Investigation (Aufgaben)

1 Baseline

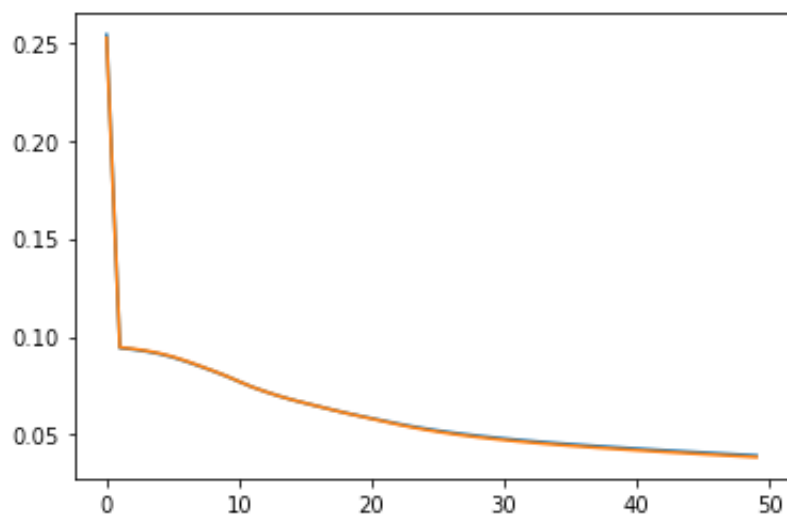
```
# initial theta-values
weights = np.random.randn(img_size_flat + 1, num_classes)
# learn rate
alpha = 0.0001
iterations = 50
```

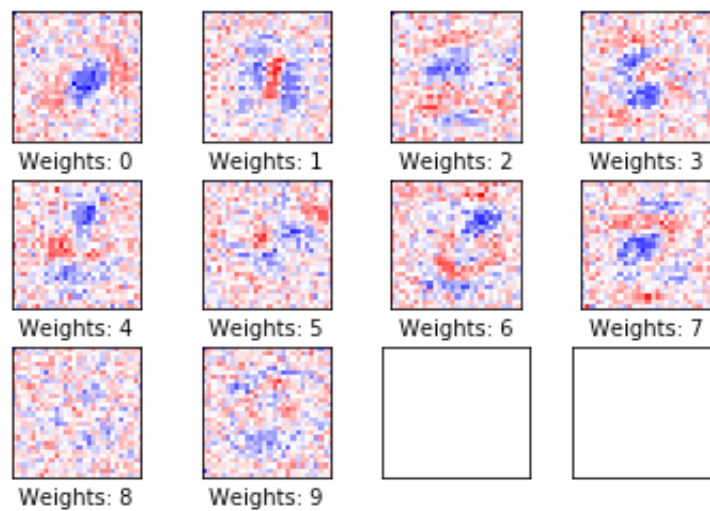
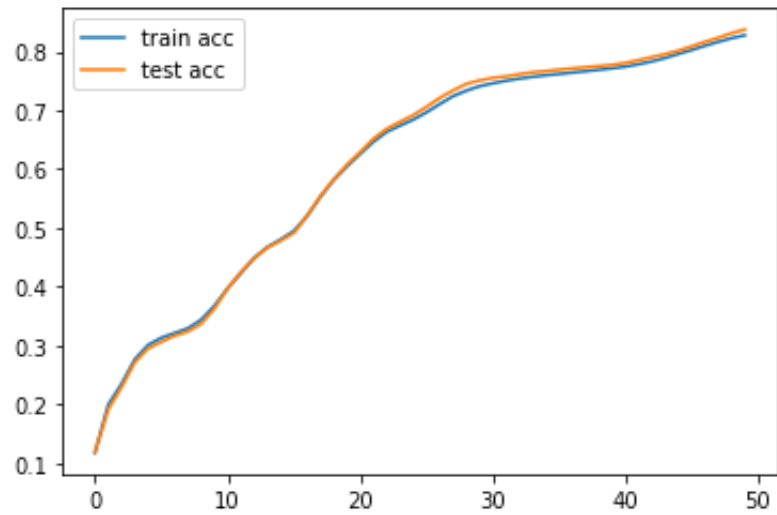




1. Gewichte initialisieren

```
# initial theta-values  
weights = np.random.randn(img_size_flat + 1, num_classes) * np.sqrt(2. /  
img_size_flat)  
alpha = 0.00001  
iterations = 50
```

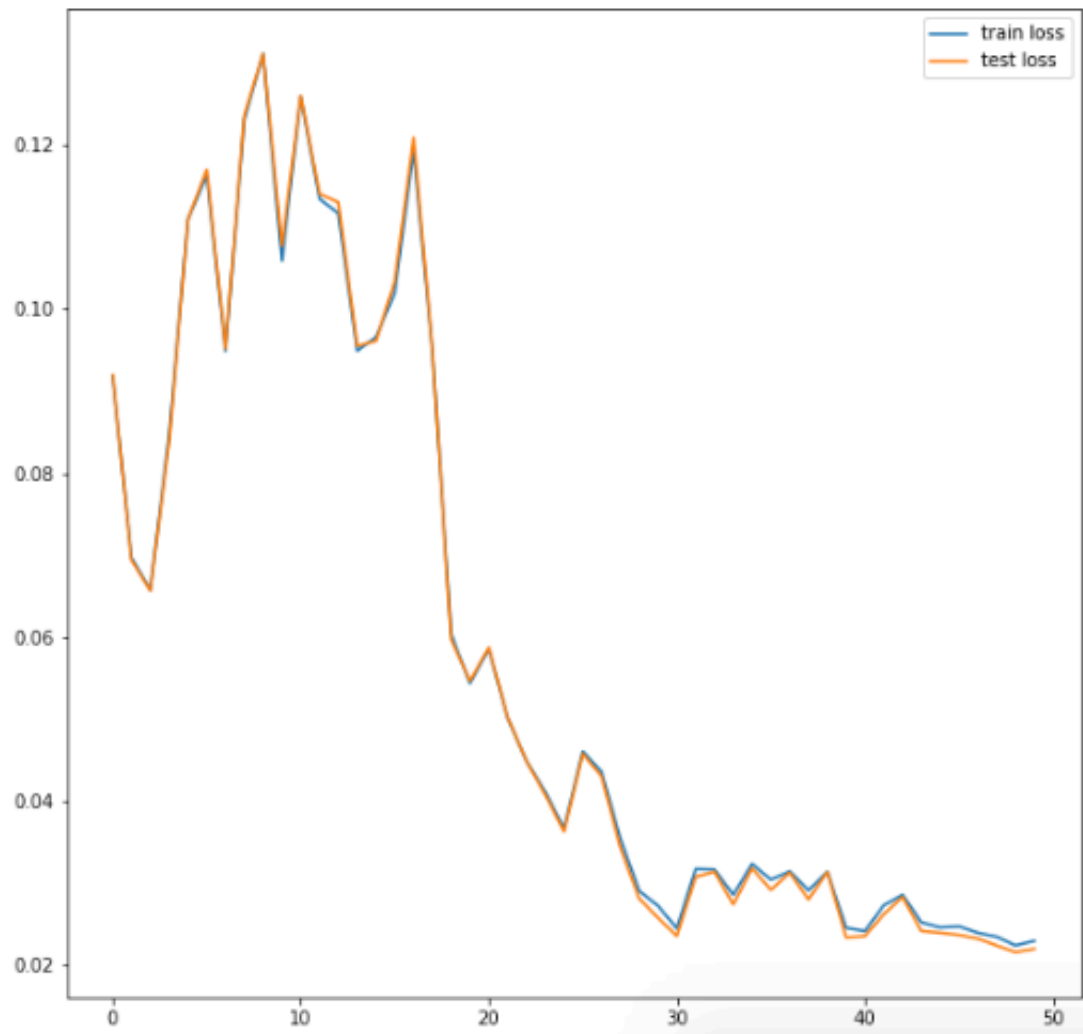


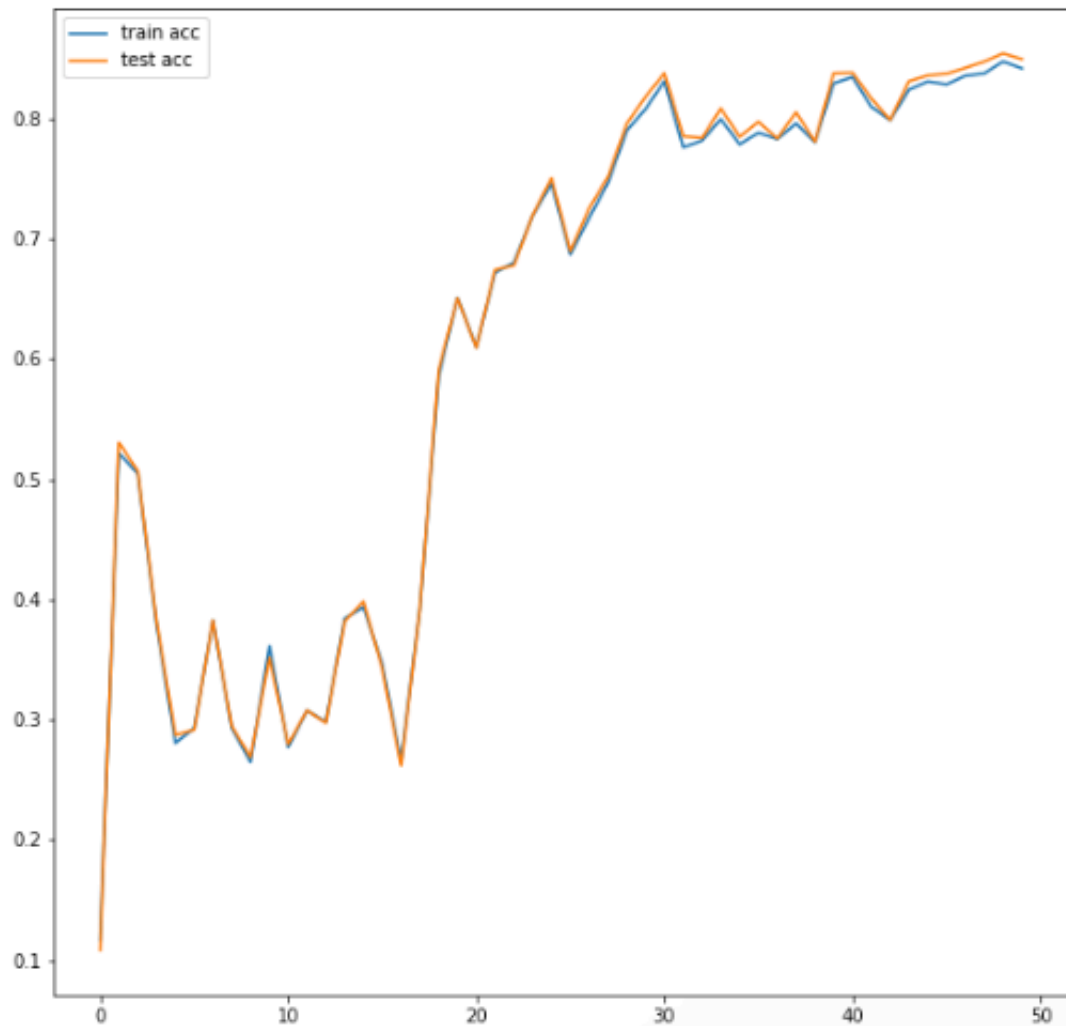


2. Aktivierungsfunktion

```
# initial theta-values
weights = np.random.randn(img_size_flat + 1, num_classes) * np.sqrt(2. /
img_size_flat)# learn rate
alpha = 0.0001
iterations = 50
```

: <matplotlib.legend.Legend at 0x7f09aa90d550>





3. Fehlerfunktion

```
# initial theta-values
weights = np.random.randn(img_size_flat + 1, num_classes) * np.sqrt(2. /
img_size_flat)# learn rate
alpha = 0.1
iterations = 50
```

