# Machine Learning - Exercise 4 implementation details:

omri fridental, gal politzer
323869545, 212257729

Structure of the network:

```
elf.conv1 = nn.Conv2d(in_channels=1, out_channels=6, kernel_size=5)
        self.conv2 = nn.Conv2d(in_channels=6, out_channels=12,
kernel_size=5)

        self.fc1 = nn.Linear(in_features=12 * 37 * 22, out_features=120)
        self.fc2 = nn.Linear(in_features=120, out_features=60)
        self.out = nn.Linear(in_features=60, out_features=30)
```

2 convolution layers:

conv1 from 1 to 6 channels, with kernel 5 * 5,
conv2 from 6 to 12 channels with kernel 5 * 5.

followed by linear layers (fully connected), with 2 hidden layers:

input layer - 12 (# out channels from conv2) * 37 * 22,
hidden layer1 - 120 neurons
hidden layer2 - 60 neurons
out layer 30 (# classes) neurons.

learning rate = 0.01
not to small and not too big, works great.

number of epochs = 20
error rate starts converge at 1 - 3 % on traininng at epoch 5, and slowly converge and upgrade accuracy.

activation function - ReLU: great function.
loss function - Cross entropy loss , like we learned that works great with relu and log softmax.