

RBM Classifier for MNIST

ISPR - Midterm 2
Assignment 3

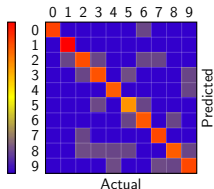
Filippo Baroni

May 3, 2021

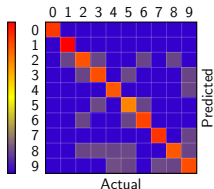
Results

RBM Classifier

Test
accuracy:
92.9%

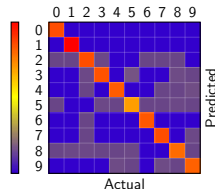


Train
accuracy:
92.7%

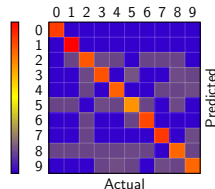


RBM + sklearn.linear_model

Test
accuracy:
90.4%

























Train
accuracy:
89.6%



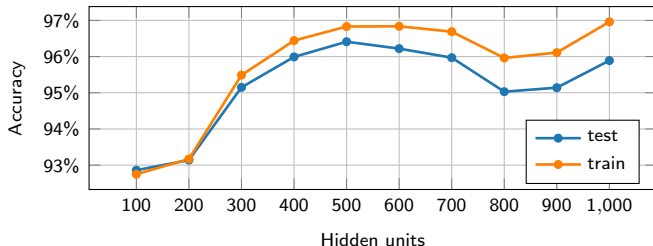
 **Neuron activations**

Experiments

► Gibbs sampling

Input	Outputs (20 iterations)									
										
										

► More hidden units



Appendix - Code

One epoch of training

```
# [...] initialize x0 and y0
h0P = sigmoid(self.c + self.W @ x0 + self.U @ y0)
wakeW = h0P @ x0.T
wakeU = h0P @ y0.T
h0 = sample(h0P)
x1 = sample(sigmoid(self.b + self.W.T @ h0))
y1 = sample(sigmoid(self.d + self.U.T @ h0))
h1P = sigmoid(self.c + self.W @ x1 + self.U @ y1)
dreamW = h1P @ x1.T
dreamU = h1P @ y1.T
# [...] perform gradient descent
```

Prediction

```
x = data.T
t = (self.c + self.U)[: , :, None] + (self.W @ x)[: , None, :]
P = np.exp(self.d) * np.product(1 + np.exp(t), axis = 0)
return np.argmax(P, axis = 0)
```