Optimization and learning, biologically inspired

Ben, Deyan, Dýrleif, Jaswanth, & Joel

What we wanted to do

Gradient Descent (GD) not biologically plausible —> What is?

Can we make an algorithm perform a task better with biologically inspired learning and optimization?

To compare across learning algorithms, the tasks we chose are:

1. PCA (Unsupervised linear model) (784 -> 100)
2. Autoencoder (Self-supervised nonlinear) (784 -> 100 -> 10 -> 100 -> 784)
3. Classification (Supervised nonlinear model) (784 -> 100 -> 10)

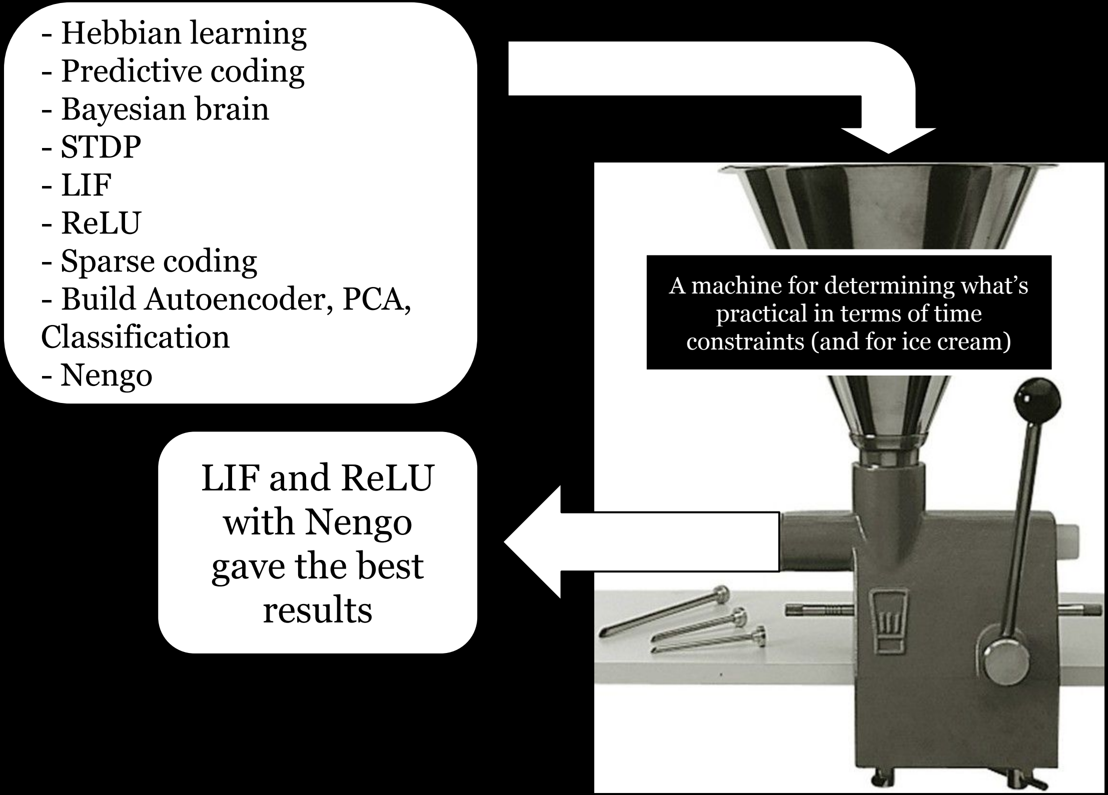
We tried to implement GD and other for these models.

What we tried and what we found

1. We have tried Predictive Coding (PC) learning algorithm to compare with GD.
2. Loss functions:
   a. MSE loss(input, output) for the Autoencoder in both settings.
      i. Tried to add MSE losses in the hidden layers for PC, but didn't help.w
      ii. The PC AE model was not learning, the loss was not decreasing. Therefore, tried a correlation loss (looked similar to the PC model architecture we were using) in the place of MSE loss. But it also didn't help.
   b. NLL loss for the Classification task model.
3. Added L1 (Lasso Regression).
4. Tried dropouts, didn't help.
5. In the end, we realised that the loss function is not helping the model and it needs to be changed.

What we ended up doing

1. Dataset: MNIST
2. PC Classification network accuracy was as bad as random classification.
3. Nengo gave the best results.
4. Its accuracies are:
5. ReLU:
   a. Accuracy before training: 8.46%
   b. Accuracy after training: 98.63%
6. LIF:
   a. Accuracy before training: 9.82%
   b. Accuracy after training: 98.7%



- Hebbian learning
- Predictive coding
- Bayesian brain
- STDP
- LIF
- ReLU
- Sparse coding
- Build Autoencoder, PCA, Classification
- Nengo

A machine for determining what's practical in terms of time constraints (and for ice cream)

LIF and ReLU with Nengo gave the best results