PyTorch libraries for linear algebra, optimization, and control

Brandon Amos
Carnegie Mellon University
http://bamos.github.io
brandondamos

block: Convenience functions for linear algebra

http://github.com/bamos/block

Let's try to construct the KKT matrix in numpy and PyTorch.

Without block, there is no way to infer the appropriate sizes of the zero and identity matrix blocks. It is an inconvenience to think about what size these matrices should be.

$$K = \left[\begin{array}{cc|cc} Q & 0 & G^T & A^T \\ 0 & S^{-1}Z & I & 0 \\ \hline G & I & 0 & 0 \\ A & 0 & 0 & 0 \end{array}\right]$$

```
K = np.bmat((
    (Q, np.zeros((nx,nineq)), G.T, A.T),
    (np.zeros((nineq,nx)), D, np.eye(nineq), np.zeros((nineq,neq))),
    (G, np.eye(nineq), np.zeros((nineq,nineq+neq))),
    (A, np.zeros((neq, nineq+nineq+neq)))
))
```

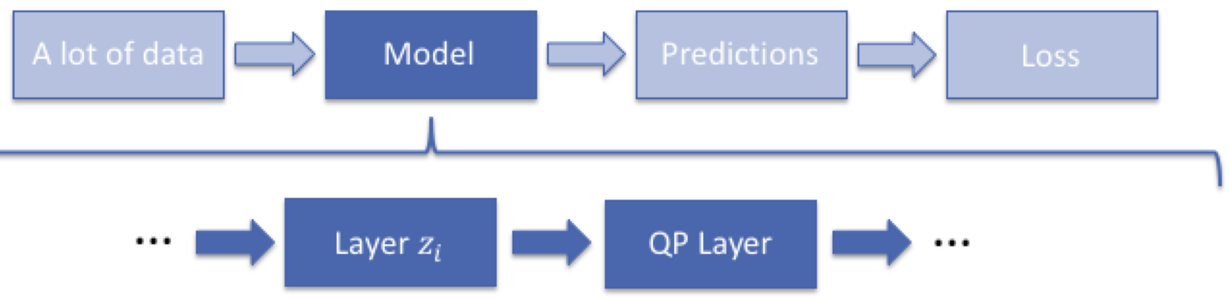
```
K = block((
    (Q,   0, G.T, A.T),
    (0,   D, 'I',   0),
    (G, 'I',   0,   0),
    (A,   0,   0,   0)
))
```

```
K = torch.cat((
    torch.cat((Q, torch.zeros(nx, nineq).type_as(Q), G.t(), A.t()), 1),
    torch.cat((torch.zeros(nineq, nx).type_as(Q), D, torch.eye(nineq).type_as(Q),
               torch.zeros(nineq,neq).type_as(Q)), 1),
    torch.cat((G, torch.eye(nineq).type_as(Q), torch.zeros(nineq,nineq+neq).type_as(Q)), 1),
    torch.cat((A, torch.zeros((neq, nineq+nineq+neq))), 1)
))
```

```
K = block((
    (Q,   0, G.t(), A.t()),
    (0,   D,   'I',   0),
    (G, 'I',   0,   0),
    (A,   0,   0,   0)
))
```

qpth: A differentiable QP layer

http://locuslab.github.io/qpth



$$z_{i+1} = \underset{z}{\text{argmin}} \frac{1}{2} z^T Q(z_i) z + q(z_i)^T z$$
$$\text{subject to } A(z_i)z = b(z_i)$$
$$G(z_i)z \leq h(z_i)$$

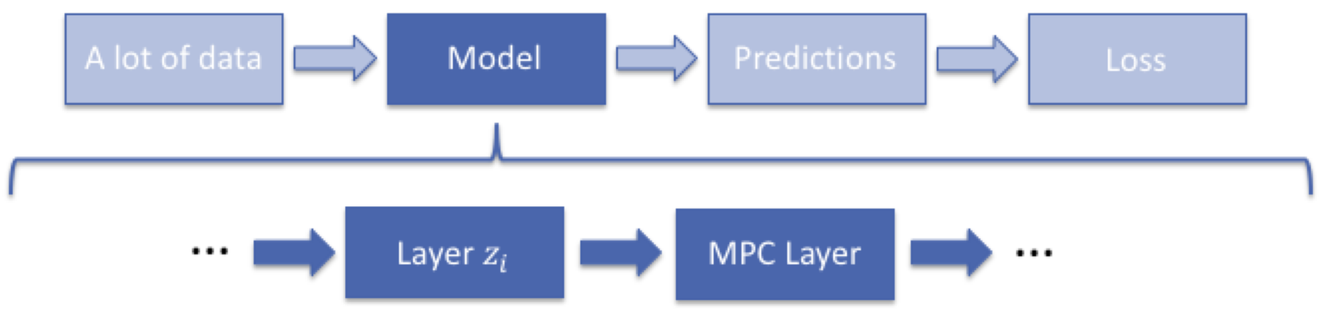
Learnable parameters: Q, q, A, b, G, h

The matrix $Q(z_i)$ depends on the previous layer z_i

OptNet: Differentiable Optimization as a Layer in Neural Networks
B. Amos and J. Z. Kolter
ICML 2017

mpc: A model predictive control layer

http://github.com/locuslab/mpc.pytorch



$$\tau_{1:T}^{\star} = \underset{\tau_{1:T}}{\text{argmin}} \sum_t C_\theta(\tau_t) \quad \text{Cost}$$
$$\text{subject to } x_1 = x_{\text{init}}$$
$$x_{t+1} = f_\theta(\tau_t) \quad \text{Dynamics}$$
$$\underline{u} \leq u \leq \overline{u}$$

Differentiable MPC for End-to-end Planning and Control
B. Amos, I. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter
NIPS 2018

Bonus

setGPU: Auto-set CUDA_VISIBLE_DEVICES

http://github.com/bamos/setGPU

A DenseNet implementation

http://github.com/bamos/densenet.pytorch