# Deep Learning as Optimal Control Problems

## 1 Introduction

Minimize cost function

$$\frac{1}{2}\sum_{i=1}^{m} |\mathcal{C}\Big(W y_i^{[N]} + \mu\Big) - c_i|^2 + \mathcal{R}(u),$$

subject to the constraint

$$y_i^{[j+1]} = y_i^{[j]} + \Delta t f(y_i^{[j]}, u^{[j]}), \qquad j = 0, ..., N-1, \qquad y_i^{[0]} = x_i$$

#### Deep Learning as an Optimal Control Problem

Typical methods solve deep learning problems by using a **first-discretize-then-optimize** approach, discretizing using a forward Euler method, and yield an optimization problem which can be solved with gradient descent. Note that Euler's method could be replaced with a more accurate integration method, but the backpropagation for computing the gradients will typically be more complicated.

The paper proposes a **first-optimize-then-discretize** approach for deriving new algorithms. This involves a *two-point boundary value Hamiltonian problem* which expresses the first order optimality conditions. The boundary value problem is solved using a numerical integration method. Naturally, a Runge-Kutta method is used forward in time, while a matching Runge Kutta method is used backwards. *If the matching Runge Kutta method is symplectic partitioned, the method is equivalent to the first, but more efficient to compute.*

## 2 Properties of the Optimal Control Problem

### 2.1 The Variational Equation

Simplifying the cost function by removing the regularization term $\mathcal{R}(u)$, the summation over all data points, and the dependency on $W$ and $\mu$ results in a simpler optimal control problem:

$$\min_{y,u} \mathcal{J}(y(T))$$

subject to the ODE constraint

$$\dot{y} = f(y, u), \quad y(0) = x.$$

The variational equation then reads

$$\frac{\mathrm{d}}{\mathrm{d}t} v = \partial_y f(y(t, u(t))v + \partial_u f(y(t), u(t))w,$$

where $\partial_v f$ is the Jacobian of $f$ with respect to $v$.

### 2.2 The Adjoint Equation

The adjoint of (10) is a system of ODEs for a variable $p(t)$ obtained assuming