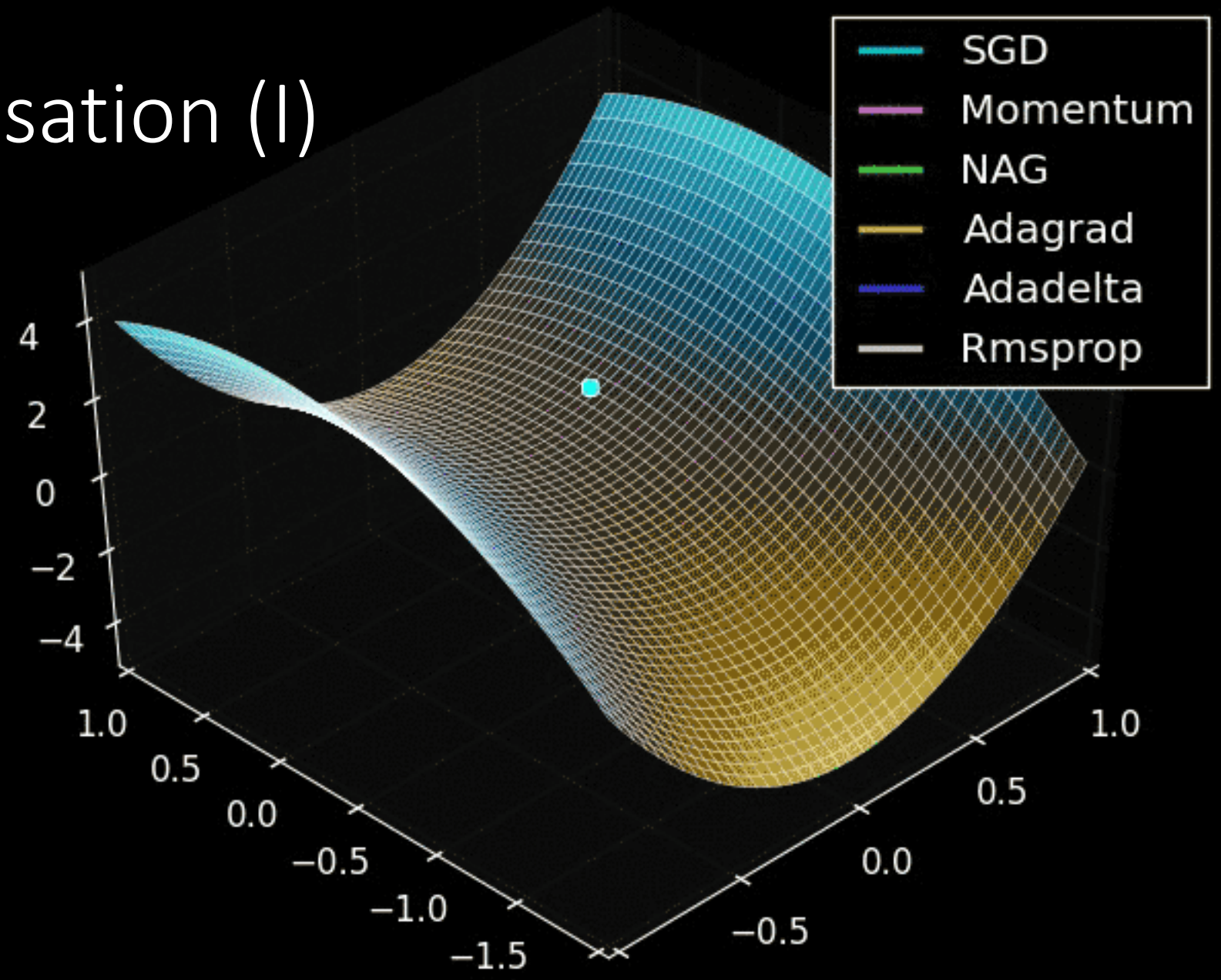
Practical 3.3

Convolutional Neural Networks – Training with `optim` package

Overview

- Non-convexity optimisation
- The `optim` package and advanced optimisers
- Training a network as obj-func optimisation
- Basic optimisation equations
- `optim` training workflow

Optimisation (I)

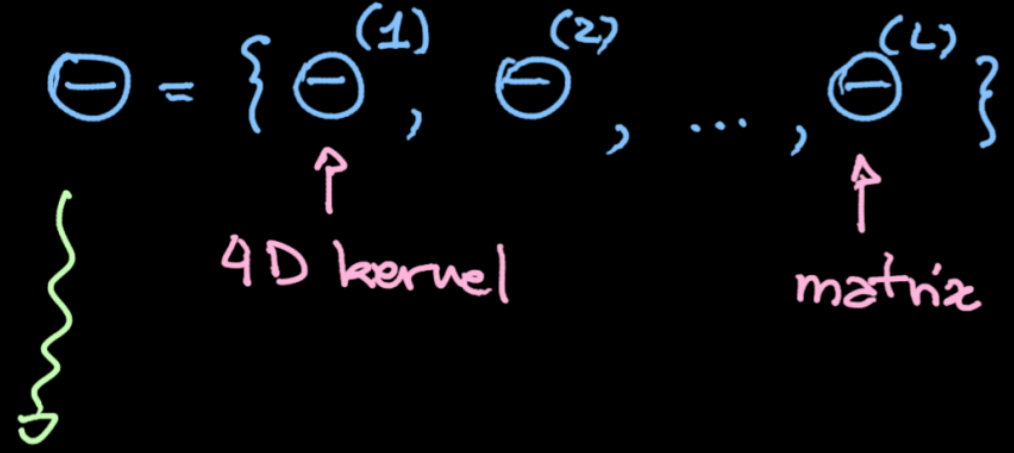


Alec Radford

Optimisation (II)

Back-propagation

$\nabla_{\underline{\theta}} J(\underline{\theta})$



$\underline{\theta} \in \mathbb{R}^d$, d: # trainable parameters

$\mathbb{R}^d$: parameter space

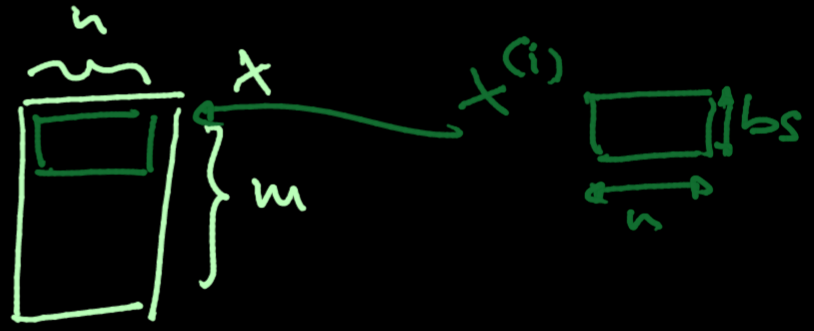
Optimisation (III)

GD $\quad \underline{\theta} \to \underline{\theta} - \eta \nabla_{\underline{\theta}} J(\underline{\theta}) = \underline{\theta} - \eta \nabla_{\underline{\theta}} J(\underline{\theta}, X, Y)$

SGD $\quad \underline{\theta} \to \underline{\theta} - \eta \nabla_{\underline{\theta}} J(\underline{\theta}, \underline{x}^{(i)}, \underline{y}^{(i)})$

MINI-BATCH GD

$$\underline{\theta} \to \underline{\theta} - \eta \nabla_{\underline{\theta}} J(\underline{\theta}, X^{(i)}, Y^{(i)})$$



MOMENTUM

$$\underline{v} \to \gamma \underline{v} + \eta \nabla_{\underline{\theta}} J(\underline{\theta}) \;, \quad \underline{\theta} \to \underline{\theta} - \underline{v}$$

Training with optim